

CSE102 - Computer Programming with C (Spring 2020)

Term Project PART-2

Vector Graphics with EPS

Handed out: 9:00 pm May 19, 2020.

Due: 23:55 pm, June 19, 2020.

Hand-in Policy: Hand in via Moodle. No late submissions will be accepted.

Collaboration Policy: No collaboration is permitted.

Grading: This project may contribute up to 5 points towards your final grade out of 100. Second part will introduce another five point.

Summary: Understanding repeating patterns: <https://www.youtube.com/watch?v=pg1NpMmPv48>

While the video in the link is a very good description of the traditional geometric art, we will employ a much simpler approach. *Which is called Hankin's method.*

In this method, like the video above, we select the geometry of the underlying grid which we will refer as **archetype**. Archetypes, may include Squares(4-Side), Pentagons(5-side), Hexagons(6-Side) etc... However You are expected to use only Squares and Hexagons while implementing this assignment.

For each side, we take the center of the edge as **contact point**. And we draw two rays pointing inside of the **tile**. The angle between rays and the origin side is called **contact angle** which is shown in the figure 1(a). This construction will give rays 2 times the number of sides. Then we will use another property that we will refer as **offset** which describes distance between the **contact point** and start of the ray figure 1(b). Then we will connect each of the by calculating their intersection points and produce the indivisible substructure of our pattern called motif. Applying this motif to our canvas repeatedly will create our final pattern.

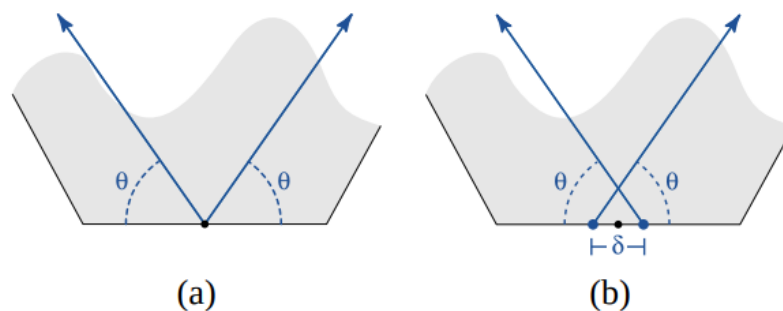


Figure 1 a) Shows the contact angle between two construction ray and b) shows the offset between center of the side and the start of the ray

Zip file includes an academic study on the subject, please read it carefully even though, we will give a brief explanation on the subject. We have also included a javascript demo file. You can execute the file by opening **index.html** file and play with the numbers and the fields to see the results.

In this part we want you to develop a software program that will generate the patterns described in a text file called **commands.txt**.

A command will define structure and visual properties of a pattern.

For ex:

```
background_color:(128,128,128),foreground_color:(255,0,0), tile_size:100,angle_offset:
(50,10), canvas_size:(1000,1000), show_grid:True, archetype:Square, file_name:sq1.eps
```

Command above is telling that our software will generate a repeating pattern in square tiles with line angle of 50 and offset of 10px. Each tile cell has an edge length of 100px. `background_color` property sets the background color of the canvas and consists of three integers between 0 and 255. Similar to `background_color`, `foreground_color` describes the color of the pattern itself. `show_grid` value tells the program that generated image whether to show the underlying grid structure or not. This command file can include different commands that you have implemented in the first part of this term project. If the property is not defined then you will fall back to the default values, described in a file called **defaults.txt**. However, there are two mandatory fields, **archetype** and **file_name**. If these properties are not described, you will produce an error giving the errors line(command) number with an appropriate error message, and **continue executing rest of the commands file**.

Commands file will include one or more commands for each pattern description you will generate a new pattern and save it to the described file. Number of commands in the file is arbitrary. You should handle the commands as a dynamic data structure.

You can find the example definitions and commands in the attachments.

Rules

1. You should use dynamic memory allocation and data structures for defining geometric objects.
2. Separation of concerns: You should split your code files in to necessary **header** and **implementation** files.
3. You should build your own makefiles to compile and build necessary source files.

What to hand in: You are expected to hand in all your source code (library and test programs) along with your makefile in a ZIP or similarly archived file named **"cse102project_lastname_firstname_studentno.zip"**. When the makefile is run, it should compile everything and produce a test program. The test program should illustrate all the above functionality.

Bağlantıdaki video, geleneksel geometrik sanatın çok iyi bir açıklaması olsa da, çok daha basit bir yaklaşım kullanacağız. Hankin yöntemi denir. Bu yöntemde, yukarıdaki video gibi, arketip olarak bahsedeceğimiz temel ızgara geometrisini seçiyoruz. Arketipler, Kareler, Beşgenler, Altıgenler vb. İçerebilir. Ancak bu ödevi uygularken yalnızca Kareler ve Altıgenler kullanmanız beklenir. Her bir taraf için, kenarın merkezini temas noktası olarak alıyoruz. Ve kiremitin içine işaret eden iki ışın çizeriz. Işıklar ve başlangıç tarafı arasındaki açığa şekil 1'de gösterilen temas açısı denir. Bu yapı ışıklara kenar sayısının 2 katı verecektir. Ardından, temas noktası ile ışın şekil 1'in başlangıcı arasındaki mesafeyi tanımlayan ofset olarak bahsedeceğimiz başka bir özellik kullanacağız. Daha sonra kesişme noktalarını hesaplayarak her birini birbirine bağlayacağız ve motif adı verilen modelimizin bölünmez alt yapısını üreteceğiz. Bu motifi tuvalimize tekrar tekrar uygulamak son modelimizi yaratacaktır.

Zip dosyası konuyla ilgili akademik bir çalışma içeriyor, lütfen dikkatlice okuyunuz, konuyla ilgili kısa bir açıklama yapacağız. Biz de bir javascript demo dahil ettik Dosya. Dizin açarak dosyayı çalıştırabilirsiniz.html dosyası ve sayıları ile oynamak ve sonuçları görmek için alanlar. Bu bölümde desenleri üretecek bir yazılım programı geliştirmenizi istiyoruz komutları adlı bir metin dosyasında açıklanmıştır.txt.

Bir komut, bir desenin yapısını ve görsel özelliklerini tanımlayacaktır.

For ex: background_color: (128,128,128), foreground_color: (255,0,0), tile_size: 100, angle_offset: (50,10), canvas_size: (1000,1000), show_grid:True, arketip:Kare, file_name:sq1.EP epss

Yukarıdaki komut, yazılımımızın kare içinde yinelenen bir desen oluşturacağını söylüyor 50 çizgi açısı ve 10px ofset ile fayans. Her kiremit hücresinin kenar uzunluğu 100 pikseldir.

background_color özelliği tuvalin arka plan rengini ayarlar ve üçten oluşur

0 ile 255 arasındaki tamsayılar. Background_color benzer, foreground_color açıklar

desenin kendisinin rengi. show_grid değeri görüntü oluşturulan programı söyler

altta yatan ızgara yapısını gösterip göstermeyeceği. Bu komut dosyası şunları içerebilir

bu terim projesinin ilk bölümünde uyguladığınız farklı komutlar. Eğer...

özellik tanımlanmadı, ardından bir dosyada açıklanan varsayılan değerlere geri döneceksiniz

varsayılan olarak adlandırılır.txt. Ancak, iki Zorunlu alan vardır, arketip ve

file_name. Bu özellikler açıklanmazsa, bir hata verir

hata satırı (komut) numarasını uygun bir hata mesajı ile ve devam

komut dosyasının geri kalanını yürütme.

Komutlar dosyası, her desen açıklaması için bir veya daha fazla komut içerecektir

yeni bir desen oluşturun ve açıklanan dosyaya kaydedin. Dosyadaki komut sayısı

keyfi. Komutları dinamik bir veri yapısı olarak ele almalısınız.

Örnek tanımları ve komutları eklerde bulabilirsiniz.

Kural

1. Tanımlamak için dinamik bellek ayırma ve veri yapılarını kullanmalısınız geometrik nesneler.

2. Endişelerin ayrılması: kod dosyalarınızı gerekli başlığa bölmelisiniz ve uygulama dosyaları.

3. Gerekli kaynak dosyaları derlemek ve oluşturmak için kendi makefiles oluşturmalsınız.